

EPICS ACCELERATOR CONTROL SYSTEMS FOR LAMBDA EMS AND TCR MAGNET POWER SUPPLIES

Anthony Andrews^{1,2*}, Y. Kim^{1,2,3}, C. Eckman^{1,2}

P. Buaphad^{1,2}, C. O'Neill², B. Berls², K. Folkman², and J. Ralph²

¹Department of Physics, Idaho State University, Pocatello, ID 83209, USA

²Idaho Accelerator Center, Idaho State University, Pocatello, ID 83201, USA

³Thomas Jefferson National Accelerator Facility, Newport News, VA 23606, USA

Abstract

The Idaho Accelerator Center (IAC) of Idaho State University has been operating a 44 MeV L-band linac for high average power THz generation, various nuclear physics related applications, medical isotope generation, and Laser Compton Scattering (LCS) based X-ray generation [1]. As of August 2012, the IAC has been upgrading the 44 MeV linac with a new computer based EPICS accelerator control system to replace the old analog control system. So far, this upgrade includes control of two GigE CCD cameras, and several magnet power supplies with embedded IEEE controllers [2]. However, the 44 MeV linac also has some magnet power supplies which are controlled with external RSTL controllers. This paper describes the development of the EPICS accelerator control system to control Lambda EMS and TCR magnet power supplies with these external RSTL controllers.

INTRODUCTION

To improve the reproducibility and stability of the 44 MeV linac, the previous analog control system was upgraded to a computer based control system. The control system chosen for this upgrade was the Experimental Physics and Industrial Control System (EPICS). As of August 2012, this upgrade included a control system for 2 Prosilica GigE CCD cameras, 32 TDK-Lambda ZUP power supplies, and 15 Lambda EMS power supplies with embedded controllers [2]. However, the 44 MeV linac also contains 3 Lambda EMS power supplies and one Lambda TCR power supply with external RSTL controllers. Figure 1 shows half of the control system for the magnet power supplies including Lambda EMS power supplies with embedded controllers, Lambda EMS with the external RSTL controllers, and a solitary Lambda TCR power supply with an external RSTL controller [3,4]¹. This paper describes how the control of these magnet power supplies with external RSTL controllers was accomplished. The full description of the EPICS accelerator control system for the 44 MeV linac will be published at another conference [5]. As shown in Fig. 1, the EMS power supplies and TCR power supplies were connected to their own dedicated MOXA terminal server with RS485-RS232 conversion in-

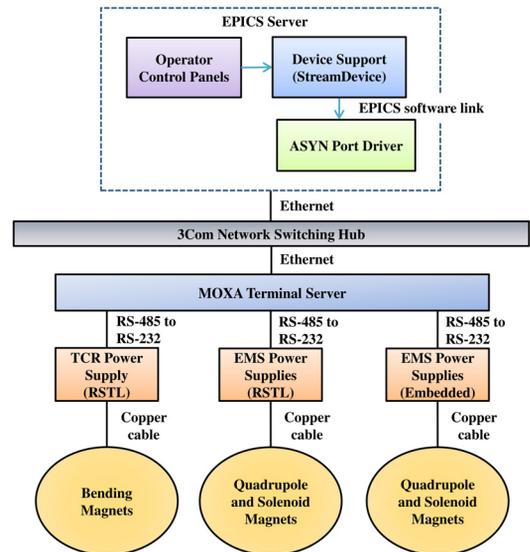


Figure 1: Experimental setup of EPICS server and hardware for the IAC 44 MeV linac.

terfaces [6]. Then, the terminal server was connected to the EPICS server via an isolated network using an Ethernet cable and a 3Com network switching hub. After that, a means to communicate with the power supplies was developed by using two EPICS modules (ASYN and StreamDevice) supplied with the synApps package [2]. Then, device support applications were manually programmed for the power supplies, and the current was controlled by manipulating process variables (PVs) using MEDM Operator Interface (OPI) panels.

HARDWARE CONFIGURATION

Before creating the PVs, the RSTL controllers had to be configured to work with the Lambda EMS and Lambda TCR power supplies. For the Lambda EMS power supplies, it was just a matter of following the connection diagram for the RSTL to TB1 connector in the manual [7]. However, the interface from the RSTL to the Lambda TCR power supply took some additional configuration for the RSTL. Since the RSTL was rated for a 40 V power supply and the Lambda TCR was a 160 V power supply, the first configuration detail was to add a couple resistors to the RSTL board. This was done to create a voltage divider and reduce the high voltage from the Lambda TCR to a level

* Mail: andranth@isu.edu

¹For more information about the internally controlled EMS power supplies see reference [2, 5]

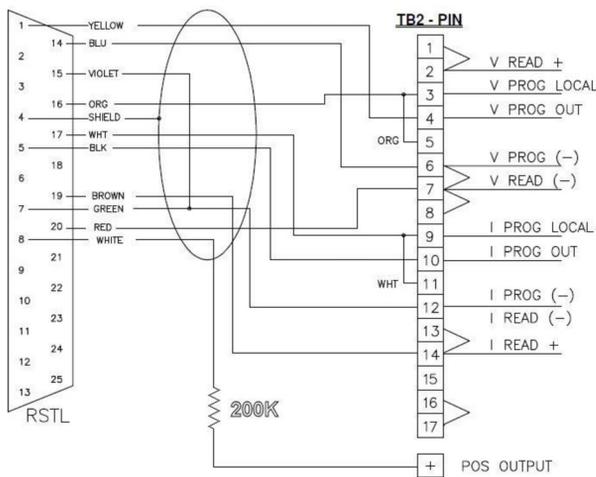


Figure 2: Connection diagram for cable going from RSTL to TB2 connector on Lambda TCR power supply [8].

that would not damage the analog to digital converter chip in the RSTL. After this, a 25 pin cable, which came with the RSTL, had to be connected from the RSTL to the TB2 connector as shown in Fig. 2 [8]. There was a 200 kΩ resistor added in series from the positive output busbar on the Lambda TCR to pin 8 on the RSTL 25 pin connector. This was done to reduce the voltage load on the 25 pin connector to within it’s specifications. To complete the hardware configuration, the Lambda TCR power supply had to be calibrated to ensure that the current readback would follow the EPICS programmed current value as close as possible [7].

DEVICE SUPPORT APPLICATIONS

After the hardware configuration and calibration, the next step was to turn off the “echo” setting of the external RSTL controller. Then make a device support application containing the files to create PVs. The device communication was accomplished by installing the synApps software package including ASYN and StreamDevice [2].

Application Creation for Power Supplies

The creation of PVs for both types of power supplies was almost the same because they both use external RSTL controllers. To create the application, the following commands were used [2]:

- `makeSupport.pl -t streamSCPI TCR_`
- `makeSupport.pl -t streamSCPI EMS_`
- `makeBaseApp.pl -t ioc EMS_and_TCR`
- `makeBaseApp.pl -t ioc -i EMS_and_TCR`

The first two commands create support directories for the Lambda EMS and Lambda TCR power supplies. These support directories contain the files to create PVs. The third command creates a configure and an application directory. The fourth command creates an iocBoot directory, which

contains the *st.cmd* file [2]. The “EMS_and_TCR” are arbitrary names for the application directory and the directory under iocBoot containing the *st.cmd* file.

Protocol Files

The module StreamDevice uses protocol files to translate commands that follow the SCPI (Standard Commands for Programmable Instruments) standard into a format that EPICS records can use [2]. For example, the SCPI command to measure the current (MC) from the RSTL manual looks like this [7]:

- `getC{out "MC"; in "%6F"; in "";}`

In this case, `getC` is a protocol, which sends the command MC “out” to the device and gets a response “in” from the device. The RSTL sends back two “in” responses, the first is a number with six floating digits, and the second is just a blank line that follows a command sent to and from the RSTL. The blank line serves as a form of confirmation that the RSTL received a command from the user. The other key component for making a PV is a record in a database file.²

Database File

The database files are where the records are located. The PVs are made by putting the protocol name (i.e. `getC`) into the input or output fields in the record [2]. Here is an example of a record to read the current:

```
record( ai, "$(P):90A_BENDS:curr_") {
    field( DESC, "Readback Current")
    field( DTYP, "stream")
    field( INP, "@TCR.proto getC $(PORT19) $(A)")
    field( PREC, "3")
    field( EGU, "Amps")
    field( HOPR, "65")
    field( LOPR, "0")}
```

In this case, the only difference between the Lambda EMS and the Lambda TCR is the definition of the High Operational Range (HOPR) of the current. For the Lambda EMS this is 30 A and for the Lambda TCR it’s 65 A. In the record above, a field of particular interest is the INP field. This field sends the command defined as the protocol “`getC`” located in the protocol file “TCR.proto” to the port and address of the device. The `$(PORT19)`, `$(A)`, and `$(P)` are all defined in the *st.cmd* file.

st.cmd File

This is an executable file that creates a channel between the RSTL and EPICS by using an IP address, a portName, and an address. This file also loads all of the records that are in the database files. An IP address is specified by entering the following line in the *st.cmd* file:

²For more protocol file information and more on SCPI commands, see reference [2].

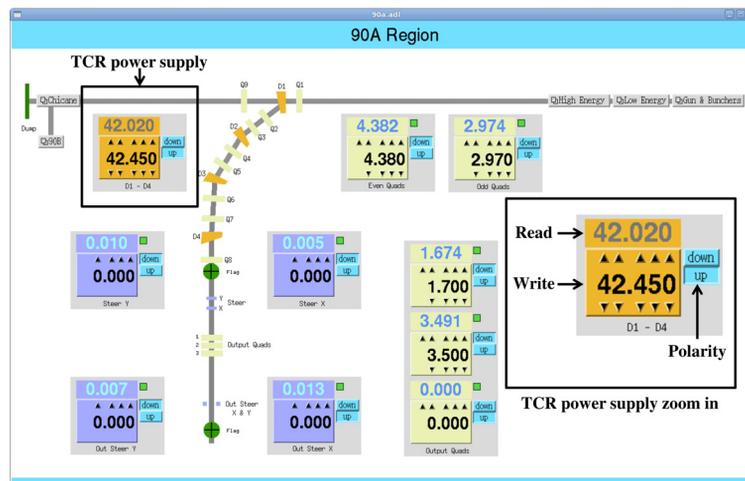


Figure 3: An MEDM OPI panel to control and read the current of the magnet power supplies for the 44 MeV linac.

```
drvAsynIPPortConfigure("P19","134.50.x.y:4019",0,0,0)
```

Here P19 is a portName, which is an arbitrary identifier used to define \$(PORT19) from records as an IP address (134.50.x.y) and a port (4019) on the MOXA terminal server [2]. In this case, P19 was used because this particular power supply was connected to the nineteenth port of the MOXA terminal server. The rest (0,0,0) specifies *priority*, *noAutoConnect*, and *noProcessEos* [2]. To specify a database in the *st.cmd* file, a line similar to the following was used:

```
dbLoadRecords("db/A.db","P=$(P),PORT19=P19,A=19")
```

Here the A.db is the database file that contains all of the needed records to control the power supply. The macro P is used as a linac identifier, in this case P is 44MeV. The PORT19 is the same as the portName. The address (A=19) for this set-up ran from 1 - 32, because there were 32 ports on each MOXA terminal server [2].

MEDM DISPLAY

Motif Editor and Display Manager (MEDM) is an EPICS extension which provides a way to create simple OPI's like the one shown in Fig. 3. This display contains a representation of the linac with the magnets being controlled. The circled box controls the TCR power supply, which controls all dipole magnets in this region of the linac. The operator changes the set (Write) current PV by clicking the black arrows or by typing a value manually. The actual readback value from the power supply is displayed in the box above write. Since all the power supplies are unipolar, the direction of the current has to be changed with external circuitry. The "Polarity" on the display gives the operator an indication of the direction of the current.³

³The other control widgets on this display are for embedded controlled power supplies, see reference [2].

CONCLUSION

The goal of this part of the 44 MeV control system upgrade was to control old Lambda EMS and TCR magnet power supplies with external RSTL controllers using EPICS. This was accomplished by using ASYN and StreamDevice to create a communication channel between EPICS and the power supplies, and then convert SCPI commands into PVs. Afterwards, MEDM OPIs were developed to control and monitor those PVs. The addition of the control system for these older magnet power supplies almost completes the EPICS upgrade for the 44 MeV linac, the rest of the control system with save/restore functionality will be described in the next conference paper [5].

ACKNOWLEDGMENTS

I would like to thank Jeff Bush, the technical group leader at TDK-Lambda Inc., for all of his help and advice in configuring the RSTL through email correspondence. I would also like to thank Dr. Kukhee Kim, Dr. Debbie Rogind, and Dr. Ernest Williams for teaching me a lot about developing EPICS device support applications through summer internships at SLAC national lab from 2011 to 2013.

REFERENCES

- [1] <http://www.iac.isu.edu>
- [2] A. Andrews et al., Proc. IPAC2013, THPEA061, Shanghai, China.
- [3] http://www.us.tdk-lambda.com/hp/product_html/emspower.htm
- [4] http://www.us.tdk-lambda.com/hp/product_html/tcrpower.htm
- [5] A. Andrews et al., to be published at ICALEPCS 2013.
- [6] http://www.moxa.com/product/nport_6650.htm
- [7] http://www.us.tdk-lambda.com/hp/pdfs/Product_manuals/83468004.pdf
- [8] A. Andrews, "Re: A problem with TDK Lambda ZUP power supply," message to Jeff Bush, 13 Mar. 2013, E-mail.